International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

Enhancing Security Measures in Online Social Networks: An Analytical Approach using Advanced Encryption Techniques

Prabhat Singh[1], Sushil Sharma[2]

[1,2]Computer Science Engineering, Institute of Technology and Management, Aligarh/Dr. APJ Abdul Kalam University, Lucknow, India

----------**********----------

ABSTRACT

The proliferation of Online Social Networks (OSNs) like Facebook, Twitter, and LinkedIn has revolutionized digital communication, enabling real-time interactions and global connectivity. However, the surge in user engagement has also made OSNs prime targets for cyber threats such as phishing, malware attacks, identity theft, and insider breaches. This paper critically analyzes existing vulnerabilities within OSNs, evaluates current security frameworks, and explores the role of advanced technologies like artificial intelligence, blockchain, and end-to-end encryption in enhancing platform security. It identifies key challenges faced by users and social media companies and proposes innovative strategies to mitigate emerging threats. The findings aim to support the development of secure and resilient OSN ecosystems, promoting trust, privacy, and user safety.

Keywords: Online Social Networks (OSN), Cybersecurity, Phishing, Artificial Intelligence, Blockchain, Encryption, Data Privacy, Insider Threats, Cyber Attacks, User Awareness

INTRODUCTION

Online Social Networks (OSNs) have become fundamental to modern communication, influencing personal, professional, and commercial interactions worldwide (Smith, 2020). Platforms such as Facebook, Twitter, Instagram, and LinkedIn have enabled billions of users to share information, build connections, and engage in business activities (Anderson &Rainie, 2019). Despite the immense benefits, OSNs have increasingly become vulnerable to cybersecurity threats including identity theft, malware dissemination, phishing attacks, and data breaches (Jang-Jaccard& Nepal, 2014).

High-profile incidents like the Cambridge Analytica scandal have highlighted the pressing need for robust security frameworks (Cadwalladr& Graham-Harrison, 2018). With users entrusting sensitive personal and professional data to these platforms, the stakes for protecting digital identities have never been higher.

This research is motivated by the urgent need to understand the evolving threat landscape and to develop innovative strategies for enhancing the cybersecurity posture of OSNs. The study focuses on examining existing threats, analyzing the effectiveness of current security mechanisms, and proposing the integration of advanced technologies like artificial intelligence (AI), blockchain, and sophisticated encryption techniques to mitigate risks.

Furthermore, the study explores how governments, businesses, education, healthcare, and financial sectors leverage OSNs, highlighting both opportunities and vulnerabilities. It also addresses the challenges users face in safeguarding personal information and outlines strategic frameworks for improving security and user awareness.

By conducting a comprehensive analysis, this research aims to contribute to creating safer online environments where individuals and organizations can confidently engage without compromising their security and privacy.

1.2 Background

Social media platforms have emerged as a fundamental mode of communication in the digital age, engaging billions of users worldwide on platforms such as Facebook, Twitter, Instagram, and LinkedIn [1]. These platforms facilitate information sharing, social interaction, and business activities, significantly influencing personal and professional spheres. However, the increasing reliance on social media has concurrently led to a rise in cyber threats, posing substantial risks to user privacy and data security [2]. Cybercriminals exploit vulnerabilities within these platforms to execute phishing attacks, disseminate malware, and exfiltrate sensitive personal information.

The dynamic and rapidly evolving nature of cyber threats has underscored the urgent need for robust and adaptive security measures to safeguard social media users [3]. This research is motivated by the escalating concerns surrounding the security of social media ecosystems. As users continue to share vast amounts of personal, financial, and professional information, ensuring the integrity and confidentiality of such data has become a critical priority. High-profile incidents, such as the Cambridge Analytica data breach, have vividly illustrated the potential consequences of inadequate security mechanisms [4].

Therefore, this study seeks to systematically analyze the prevailing cyber threats associated with social media platforms and propose effective strategies to mitigate these risks, ultimately contributing to the creation of a safer and more resilient digital environment.

Problem Statement

Despite significant advancements in cybersecurity, social media platforms remain vulnerable to evolving cyber threats. Cybercriminals continually develop sophisticated techniques to exploit security vulnerabilities, steal personal data, and manipulate user information [10]. A major contributing factor is the lack of user awareness regarding cybersecurity best practices, with many falling victim to phishing attacks, identity theft, and malware infections due to weak passwords, unverified links, and careless sharing of personal details [11].

This research aims to address the following issues:

- The increasing prevalence of cyber threats on social media platforms.
- Inefficiencies and limitations of current security measures.
- The need for advanced security strategies to counter emerging threats.
- The role of artificial intelligence, blockchain, and encryption in strengthening security.

By tackling these challenges, the study seeks to provide effective solutions to mitigate cyber threats in the social media domain.

4. Scope of the Study

This study focuses on the cybersecurity landscape of social media platforms, with an emphasis on identifying key threats and corresponding countermeasures. The research covers the following areas:

- A comprehensive analysis of cyber threats targeting social media users, including phishing, malware, social engineering, and data breaches [12].
- Evaluation of current security frameworks implemented by major platforms such as Facebook, Twitter, and LinkedIn [13].
- Exploration of advanced technologies such as artificial intelligence, blockchain, and end-to-end encryption for enhancing cybersecurity [14].
- Development of new strategies to mitigate cyber risks and improve user awareness of online safety [15].

The findings will assist social media companies, cybersecurity professionals, and policymakers in developing effective security measures and protective policies for users.

RESEARCH OBJECTIVE

This study aims to develop strategies and solutions for safeguarding personal information and profiles within OSNs. Various threats and vulnerabilities associated with OSN usage highlight the need for improved security measures. To address these concerns, it is essential to design robust models and methods that enhance privacy and security within Online Social Networks

LITERATURE REVIEW

This section discusses key studies focused on password security and authentication mechanisms, exploring various techniques to enhance security on social media platforms.

- **Password Authentication**: Ziegler et al. (2014) noted that while passwords are cost-effective and widely used, they are vulnerable to theft and brute-force attacks.

- **Password Hashing**: Wang et al. (2019) and Mendelson et al. (2020) emphasized password hashing as a secure method to store passwords, using irreversible hash values to enhance security.
- **Password Breaches**: Despite hashing, breaches persist due to brute-force and dictionary attacks. High-profile cases like Yahoo and LinkedIn breaches (Schneier, 2013; Khandelwal, 2014) highlight ongoing vulnerabilities.
- **Multi-Factor Authentication (MFA)**: Studies by Qiu et al. (2020) and others show that MFA, including two-factor and three-factor authentication, improves security but increases complexity. Platforms like Twitter have adopted MFA for better protection.
- **Salting in Hashing**: Kioon et al. (2013) highlighted salting as a method to further secure passwords, ensuring that identical passwords have unique hash values, making them harder to crack.
- **Key Stretching**: Kelsey et al. (2006) introduced key stretching, which makes brute-force and dictionary attacks more difficult by increasing the computational cost through multiple encryption iterations.
- **Honeywords Mechanism**: Juels&Rivest (2013) proposed honeywords—decoy passwords used to detect unauthorized access attempts, enhancing password security by misleading attackers.
- **Honeyword Generation Models**: Akshima et al. (2019) identified three models for honeyword generation: evolving models, profile-based models, and secret code appending, each improving deception in password security.
- **Honeychecker System**: Erguler et al. (2016) developed the honeychecker system, which verifies passwords and detects unauthorized login attempts, helping to differentiate real passwords from decoys.
- **Alternative Authentication Methods**: Akshima et al. (2019) explored Encrypted Negative Passwords (ENP), which use multiple encryption layers but remain vulnerable if salts are compromised.

These studies underscore the importance of evolving and combining various security measures to mitigate cyber threats and improve user authentication mechanisms in social media platforms.

METHODOLOGY

n the research methodology you provided, **Long Short-Term Memory (LSTM)** networks have been used to detect **malicious insiders** in social networks. The key methodology used includes:

1. **Behavioral Analysis**: User activities such as login patterns, file uploads/downloads, and interactions are continuously monitored to detect anomalies in behavior. These behaviors are compared against established legitimate behavior profiles.
2. **LSTM Algorithm**: The core methodology leverages the LSTM network for analyzing time-dependent user behavior data. LSTM is a type of recurrent neural network (RNN) that is effective at identifying sequences and patterns over time, making it suitable for detecting deviations in user behavior on social networks.
3. **Anomaly Detection**: If a user's behavior deviates significantly from the norm (based on LSTM predictions), an alert is generated. These deviations are then further investigated to classify them as normal or malicious.
4. **Node Classification**: Using the trust factors computed by the **TBTE (Time-Based Trust Evaluation)** algorithm, nodes (users) are classified into:
    - **Normal nodes** (trusted users)
    - **Broken nodes** (users with little or no activity)
    - **Malicious nodes** (users exhibiting suspicious behavior)
5. **Trust Factor Analysis**: A balance factor is calculated based on time-based quantitative analysis to reduce false positives, ensuring accurate classification of nodes.

The primary methodology used in this research paper is the **LSTM-based anomaly detection system** for identifying **malicious insiders** in social networks.

To implement an LSTM-based anomaly detection system for malicious insider detection in social networks, you would use Python, Keras, TensorFlow, and tools like NumPy and Pandas. The LSTM model helps in identifying anomalies in user behavior by analyzing time-series data, and you can use cloud platforms for deployment.

HONEYWORDS: AN ADVANCED SECURITY MEASURE

Passwords remain a widely used authentication method on online platforms, but they are prone to vulnerabilities and breaches. Cyberattacks targeting password databases continue to threaten online security, necessitating enhanced protection methods. One innovative approach to strengthen password security is the use of **honeywords**, decoy passwords that serve as a trap for attackers attempting to steal credentials.

In this system, the actual user password, referred to as the **sugarword**, is stored alongside multiple **honeywords** in the password database. Collectively, these passwords are termed **sweetwords**. If an attacker attempts to log in using a honeyword, it indicates a potential breach, alerting system administrators to unauthorized access attempts.

Recent advancements in honeyword-based authentication have led to the development of sophisticated algorithms for generating realistic honeywords, reducing the likelihood of attackers distinguishing between the actual password and the decoys. Machine learning and artificial intelligence are increasingly being explored to improve honeyword generation and detection. Despite these advancements, the optimal method for securing password databases remains an area of ongoing research.

Honeywords as an Adv Key Contributions of HAaT:

1. **Enhanced Honeyword Generation**: Utilizes AI-driven algorithms to generate highly realistic honeywords that are difficult for attackers to distinguish from the real password.
2. **Adaptive Detection Mechanism**: Employs behavioral analysis to detect anomalous login attempts in real-time, indicating potential breaches.
3. **Automated Breach Response**: Integrates with security alert systems to trigger countermeasures when a honeyword is used.
4. **Improved User Authentication Security**: Enhances security without significantly increasing authentication complexity.
5. **Scalability and Integration**: Easily integrates with various online platforms without requiring significant architectural changes.

By incorporating these advancements, HAaT offers a robust solution to password security, minimizing credential theft risks and strengthening authentication mechanisms.

Overview of Honeywords as an Advanced Trap (HAaT)

HAaT utilizes an advanced **Rule-Based Honeyword Generator (RHG)** to generate honeywords for a user's password (**Pi**). To further enhance security, HAaT employs a **Salt-Chlorine Generator (SCG)** to generate complex salts for hashing both the real password and the decoy honeywords.

The **RHG** generates a set of honeywordsWi={w1,w2,...,wk−1}Wi = \{w_1, w_2, ..., w_{k-1}\}Wi={w1,w2,...,wk−1} that are indistinguishable from the real password. The **SCG** then generates unique salts for hashing, producing a final list of hashed sweetwordsh(Wi)={h(wsc1),h(wsc2),...,h(wscn)}h(Wi) = \{h(wsc_1), h(wsc_2), ..., h(wsc_n)\}h(Wi)={h(wsc1),h(wsc2),...,h(wscn)}. This multi-layered approach ensures that even if an attacker gains access to the password database, distinguishing the legitimate password from honeywords remains virtually impossible.

Authentication Process:

1. The user submits their password (**Pi**) for authentication.
2. The system checks the submitted password against the stored sweetwords.

3. If a match is found, the system forwards the index of the password to the **Honeychecker** for verification.
4. The **Honeychecker** verifies whether the submitted index corresponds to the real password.
5. If the indices match, authentication is granted.
6. If the indices do not match and correspond to a honeyword, an alert is triggered.

This system significantly enhances password security by adding an extra verification layer, making it difficult for attackers to compromise accounts using brute force or dictionary attacks.

Key Security Enhancements:

- **Rule-Based Honeyword Generation (RHG)**: Generates highly convincing decoy passwords based on predefined security rules.
- **Salt-Chlorine Generator (SCG)**: Uses cryptographic salts to harden password hashes, making them resistant to attacks.
- **Honeychecker Mechanism**: Provides an additional validation layer to detect unauthorized access.
- **Automated Alerts**: Alerts administrators when a honeyword is used, signaling a potential security breach.
- **Resistance to Attacks**: Even if the database is compromised, attackers cannot easily identify the real password.

By integrating these techniques, HAaT strengthens password security, ensuring that online systems are resilient against cyber threats and unauthorized access attempts.

Password Policy

To maintain strong security, users must follow the following password policies:

1. **Uniqueness**: Passwords must not be related to the username (e.g., if the username is "Jack," the password should not be something predictable like "sparrow").
2. **Distinct Credentials**: Users should avoid using the same password across different platforms to prevent credential-stuffing attacks.
3. **Substring Restriction**: Passwords must not contain any substring derived from the username.
4. **Length and Complexity**: Passwords must be at least 10 characters long and include a combination of:
    - Uppercase letters (A-Z)
    - Lowercase letters (a-z)
    - Numbers (0-9)
    - Special characters (!, @, #, $, etc.)
5. **Entropy Enhancement**: Passwords should be randomly generated or derived using strong passphrase techniques to prevent common password attacks.
6. **Regex Pattern Validation**: Passwords must conform to patterns like:
    - [0-9]+ (Numeric strength)
    - [a-zA-Z0-9]+ (Combination of uppercase, lowercase, and numbers)

These policies ensure robust password security by reducing predictability and increasing resistance to brute-force and dictionary attacks.

Alarm Policy

HAaT implements an adaptive **alarm policy** based on password-guessing complexity. To mitigate **Denial-of-Service (DoS)** attacks, stricter alarm policies are enforced for high-frequency login attempts. Since security risks vary across different user groups, alarm policies are customized based on system vulnerabilities and attack patterns, rather than following a uniform structure.

The number of login attempts MMM is calculated using Equation 3.1:
M=(k−N)2M = \frac{{(k - N)}}{2}M=2(k−N)

Where:

- NNN is the total number of passwords in the system.
- kkk is the number of honeywords per password.

This equation ensures that attackers attempting a DoS attack cannot easily lock out legitimate users or compromise the authentication system.

Advanced Honeyword Generation Mechanism (AHGM)

An **Advanced Honeyword Generation Mechanism (AHGM)** has been introduced in HAaT to enhance security by deploying decoy passwords, making it more difficult for attackers to differentiate real passwords from decoys. This mechanism utilizes sophisticated hashing techniques, including salt-chlorine encryption, to bolster security.

Current honeyword generation methods rely on user passwords, which could lead to vulnerabilities if attackers gain access to personal information. The AHGM addresses these vulnerabilities by using a more strategic rule-based honeyword generation process.

Honeyword Generation Strategy

The AHGM follows these rules for honeyword generation:

1. **Half of the honeywords** are generated using the user's personal information, while the other half are randomly generated.
2. Each honeyword must maintain a minimum **Levenshtein distance** from the actual password and maximize the **hamming distance** between different honeywords.
3. **No honeyword** should match any previous passwords used by the user to prevent credential reuse attacks.
4. Honeywords are updated every time the user changes their password.
5. The length of each generated honeyword must be at least equal to or greater than the user's real password.
6. Each honeyword is generated with **unique encryption seeds** to prevent pattern correlation with the original password.
7. 5.7.2 Secure Honeyword List Generation

The AHGM method generates honeywords based on user attributes and randomized passwords. The algorithm selects the top honeywords with the highest **hamming distance** from the real password, ensuring that they are distinct and difficult to guess.

Algorithm for Honeyword Generation:

Input: User Information (UIiUI_iUIi)

Output: 'k' honeywords

1. Generate words based on user information: words[i] = combination_gen(UI_i)
2. Compute the hamming distance for each word: value[i] = hamming_distance(password, words[i])
3. Generate random passwords: words1[i] = random_passgen[1, n+1]
4. Compute the hamming distance for random passwords: value1[i] = hamming_distance(password, words1[i])
5. Select the top honeywords based on distance.

By using these rules and algorithms, AHGM creates highly secure and unpredictable honeywords, making it more challenging for attackers to differentiate between real passwords and decoy passwords.

SECURITY EVALUATION

The proposed Honeywords as an Advanced Trap (HAaT) method has undergone testing against various attack vectors, showcasing its resilience compared to traditional password protection approaches.

Password Guessing Password guessing is a primary attack method, and HAaT combats two types:

- **General Password Guessing**: Traditional systems lack control over password creation, leaving passwords vulnerable to online guessing tools. HAaT mitigates this by generating honeywords that mislead attackers, triggering alerts when incorrect passwords are used.
- **Targeted Password Guessing**: Attackers leverage personal information like age and name to guess passwords. HAaT strengthens security by generating honeywords based on user-specific data, misleading attackers and ensuring that identifying the correct password remains highly improbable.

Brute Force Attack In a brute force attack, an attacker systematically tries all possible passwords. HAaT's generation of honeywords based on user data makes guessing highly challenging. Additionally, the system employs salt-chlorine encryption, ensuring that even with access to hashed sweetwords, attackers cannot easily decrypt the real password. The inclusion of a honeychecker further strengthens the defense, rendering brute force attempts ineffective.

Dictionary Attack This type of attack involves systematically testing dictionary words to crack passwords. HAaT mitigates such attacks through strict password policies and salting techniques, making it difficult for attackers to match sweetwords with dictionary entries.

Denial of Service (DoS) AttackADoS attack targets the system by overwhelming it with excessive login attempts, often using incorrect honeywords. HAaT reduces the risk by adding complexity to the honeyword generation, making it harder for attackers to predict the right one. Additionally, HAaT can handle up to 11 failed login attempts before triggering an alert, further decreasing the risk of a successful DoS attack.

DoS Attack EvaluationHAaT's resistance to DoS attacks was tested with 1,000 real-world passwords, each with 20 honeywords. The attackers were able to generate only a few matches with HAaT-generated honeywords, thanks to the added complexity. The system's alert system ensures minimal risk even in the case of repeated failed login attempts.

EXPERIMENTAL ANALYSIS

Flatness Evaluation:
The effectiveness of HAaT'shoneyword generation method was evaluated by measuring its "flatness," which ensures that the distribution of sweetwords makes it difficult for attackers to predict the correct password. The system was tested with bcrypt hashing and a salt-chlorine approach.

Brute Force Resistance:
Password-cracking simulations using the Hashcat tool showed that cracking passwords shorter than three characters was relatively easy, but as password length increased to 15–32 characters, the time required for cracking significantly increased, demonstrating stronger resistance. The salted sweetwords further increased this resistance by making decryption more difficult.

Comparison of Traditional vs. Salted Sweetwords:

- **Table 3.1** presents cracked sweetwords stored without salt. These passwords were vulnerable to attacks due to the lack of additional security layers.
- **Table 3.2** showcases the salt-chlorine values generated for passwords. These salted sweetwords are significantly harder to crack due to added encryption complexity.

Salt-Chlorine Process: The salt-chlorine method involves a multi-step transformation process to enhance password security, making it more resistant to brute force and dictionary attacks. This includes:

- PRSG (Pseudorandom Sequence Generator) for encryption,
- ASCII conversion for further encoding,

- Shuffling and filtering for additional complexity,
- Symbol transformation to obscure password patterns.

This method ensures that even if an attacker gains access to hashed sweetwords, they cannot easily decrypt the passwords. In conclusion, HAaT's advanced honeyword generation and salting techniques provide a robust defense against various types of cyber-attacks, making it a reliable solution for enhancing online authentication security.
anced Trap (HAaT) is introduced as an enhanced approach to mitigate password database theft and improve online security.

Table 1.1 Hash Cracking on Sweet words Without Salt

Hashed sweet word list	Hash type	Raw passwords (Reverted Hashes)
6e70477ad778d804d7689f53372047c44455fb8bc006 778856c231b6605186cf58ca087c373aa263909c7cfb 368a0ea8927880cc089812ee	sha512	james007
9ef74b5b454c0681f88c77a9fac29dfc	md5	HYDRA123
1cf0295b47683ca5d354af46ff977bf2de6d70e6749d3 078fbfe580f2f3395e3	sha256	Angel31
e241ea1f510611a85765e48ceaa7b55ece0bf640e	sha1	Krupskaya
191f6cf307771c3594bbe2945dbb36de	tiger128,3	SWEET789

Table 1.2 Salt-Chlorine for Passwords

Password Salt-chlorine	Password Salt-chlorine
HYDRA123 a%M)-WH!R9P	HYDRA123 a%M)-WH!R9P
. . . . .	. . . . .
Angel31	oQQL[,FeiCJ/nz>
Krupskaya	<fiGE6p2W"2e^08
SWEET789	0O)vDT+"G

Table 1.3 Salt-Chlorine Process

PRSG	ASCII conversation	Shuffling	Filtering	Symbol transformation
5WWR8snV FmujbdE	53 87 87 82 56 115 110 86 70 109 117 106 98 100 69 32	53 17 80 88 56 175 129 86 96 101 917 106 00 108 17 32	53 40 80 88 56 80 34 86 96 101 62 106 32 108 49 3	5(PX8P"V"e>j l1(space)
MOh0oz M3jFPF9Cf	77 79 104 48 111 122 77 51 106 70 80 70 57 67 102	97 05 172 41 710 182 72 01 177 57 80 60 17 67 104	97 37 77 41 45 87 72 33 82 57 80 60 54 67 104	a%M)- WH!R9P<Ch
pYQLazFtiCJfifV	112 89 81 76 97 122 70 116 105 67 74 102 105 102 86	111 81 81 76 91 012 70 196 105 67 74 807 205 122	62 111 81 81 76 91 44 70 101 105 67 74 47 110 122 62	oQQL[,FeiCJ/n z>
QfiLc6zbaBhg msv	81 102 105 76 99 54 122 98 97 66 104 103 109 115 118	60 102 105 71 69 54 112 28 87 96 145 101 189 903	911 60 102 105 71 69 54 112 50 87 96 50 101 94 48 56	<fiGE6p2W"2e ^08
8O1vDrvnGW Cdsnd	56 79 49 118 68 114 118 110 71 87 67 100 115 110 100	16 79 41 118 68 084 518 191 71 07 60 107 115 011 110	48 79 41 118 68 084 43 96 71 39 60 107 115 043 110	0O)vDT+"G<ks +o

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

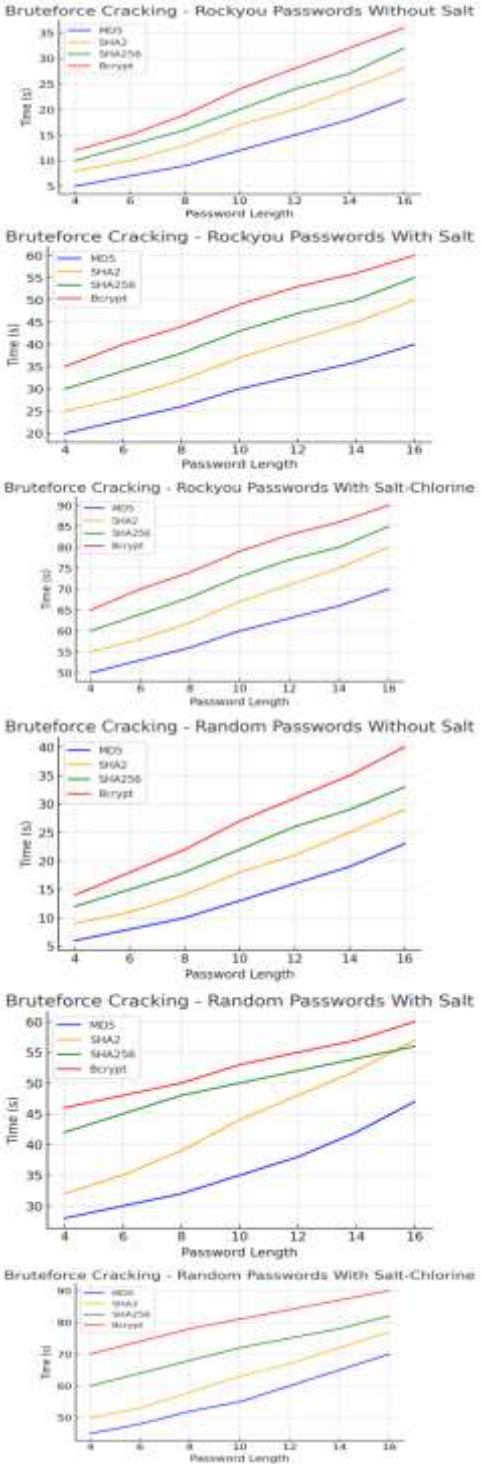
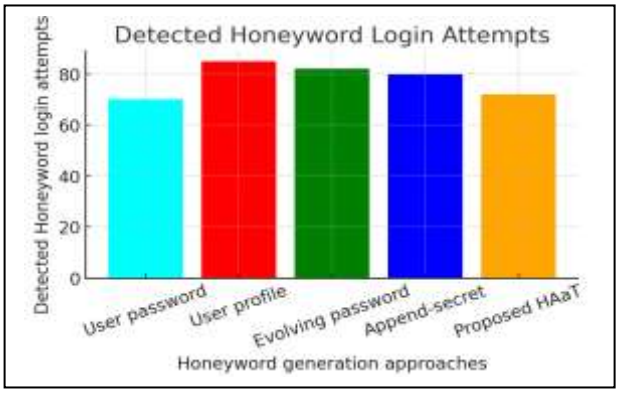

Bruteforce Cracking - Rockyou Passwords Without Salt
Bruteforce Cracking - Rockyou Passwords With Salt
Bruteforce Cracking - Rockyou Passwords With Salt-Chlorine
Bruteforce Cracking - Random Passwords Without Salt
Bruteforce Cracking - Random Passwords With Salt
Bruteforce Cracking - Random Passwords With Salt-Chlorine
MD5
SHA2
SHA256
Bcrypt
Time (s)
Password Length

Figure 1. Different combinations of Brute force password cracking



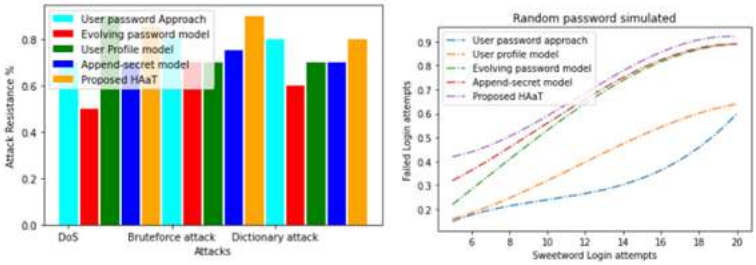


Figure 2 (a)HAaT attack resistance comparison with existing approaches, (b) Sweetword login attempts on Simulated dataset

9. Detection of Fraudulent Promotions in Online Social Networks

Online Social Networks (OSNs) such as Instagram, Twitter, Facebook, and YouTube play a crucial role in connecting individuals through shared interests and providing platforms for interactive communication. As these platforms grow in

influence, they have become attractive targets for malicious actors who exploit them by disseminating false or misleading promotional content. This deceptive activity can mislead consumers and distort market dynamics, necessitating robust detection mechanisms.

To address this pressing challenge, this research presents the Social Fraud Detection (SFD) model, a novel system designed to efficiently identify fraudulent promotional activities within OSNs. The SFD model integrates advanced URL tokenization, web crawling, and feature analysis to enhance the detection and classification of malicious content.

9.1 Overview of the Proposed Social Fraud Detection (SFD) System
The SFD system aims to systematically detect and neutralize fraudulent content embedded within social media platforms. It focuses on categorizing URLs associated with user-shared posts as either benign or malicious.

At the core of the SFD framework lies the **Deterministic Finite Automata Tokenization (DFA-T)** module, which dynamically analyzes consumer-related URLs by recognizing suspicious tokens and patterns. To complement this, an **enhanced web crawler** is employed to develop a **Web Crawler Index (WCI)** — a metric crucial for identifying newly created, potentially harmful websites.

The final decision-making process is handled by the **Web Classifier with Network and Feature Signals (WC-NFS)** module, which integrates data from DFA-T, web crawling outputs, and web traffic analyses. This cohesive architecture ensures a reliable classification of URLs and supports timely fraud mitigation.

9.2 Fraud Detection Process Using the SFD Model
The SFD model operates by analyzing URLs and associated web features through six key indicators:

- **Subdomain (SD)**
- **Path Domain (PD)**
- **Penalty Score (PS)**
- **Google Index (GI)**
- **Web Crawler Index (WCI)**
- **WHOIS Information**

The detection process is initiated by DFA Tokenization (DFA-T), wherein extracted URLs are parsed to retrieve the SD and PD. A **Penalty Score (PS)** is then computed based on the presence of suspicious keywords or blacklisted terms within the URL.

Subsequently, the **Web Traffic Analyzer (WTA)** assesses the **Google Index (GI)** of the page, determining whether the page is indexed by major search engines — a key indicator of credibility. The system further generates the **Web Crawler Index (WCI)** through dynamic crawling techniques that detect recently established malicious websites.
By evaluating these parameters collectively, the SFD model classifies URLs as either trustworthy or fraudulent. The overall system architecture, encompassing DFA-T, WTA, and WC-NFS, is visually represented in Figure 5.1.

9.3 DFA Tokenization (DFA-T)
The DFA Tokenization (DFA-T) module is activated post-URL extraction from user content. Utilizing a deterministic finite automaton, this module systematically processes URLs, dividing them into logical tokens based on predefined syntactic rules.

The standardized format used for tokenization is:
plaintext
CopyEdit
<protocol>://<subdomain>.<primarydomain>.<top-level-domain>/<pathdomain>
During tokenization, each character of the URL stream — referred to as the *lexeme* — is sequentially read and stored in an input buffer. A **Pointer Value (PV)** traverses the buffer, progressing character-by-character to construct tokens dynamically.

This methodical parsing not only accelerates URL analysis but also enhances the system's ability to detect nuanced patterns of fraudulent activity. An illustration of the DFA-T operational workflow is provided in Figure 5.2, and Table 5.1 showcases a sample structural breakdown of a URL.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

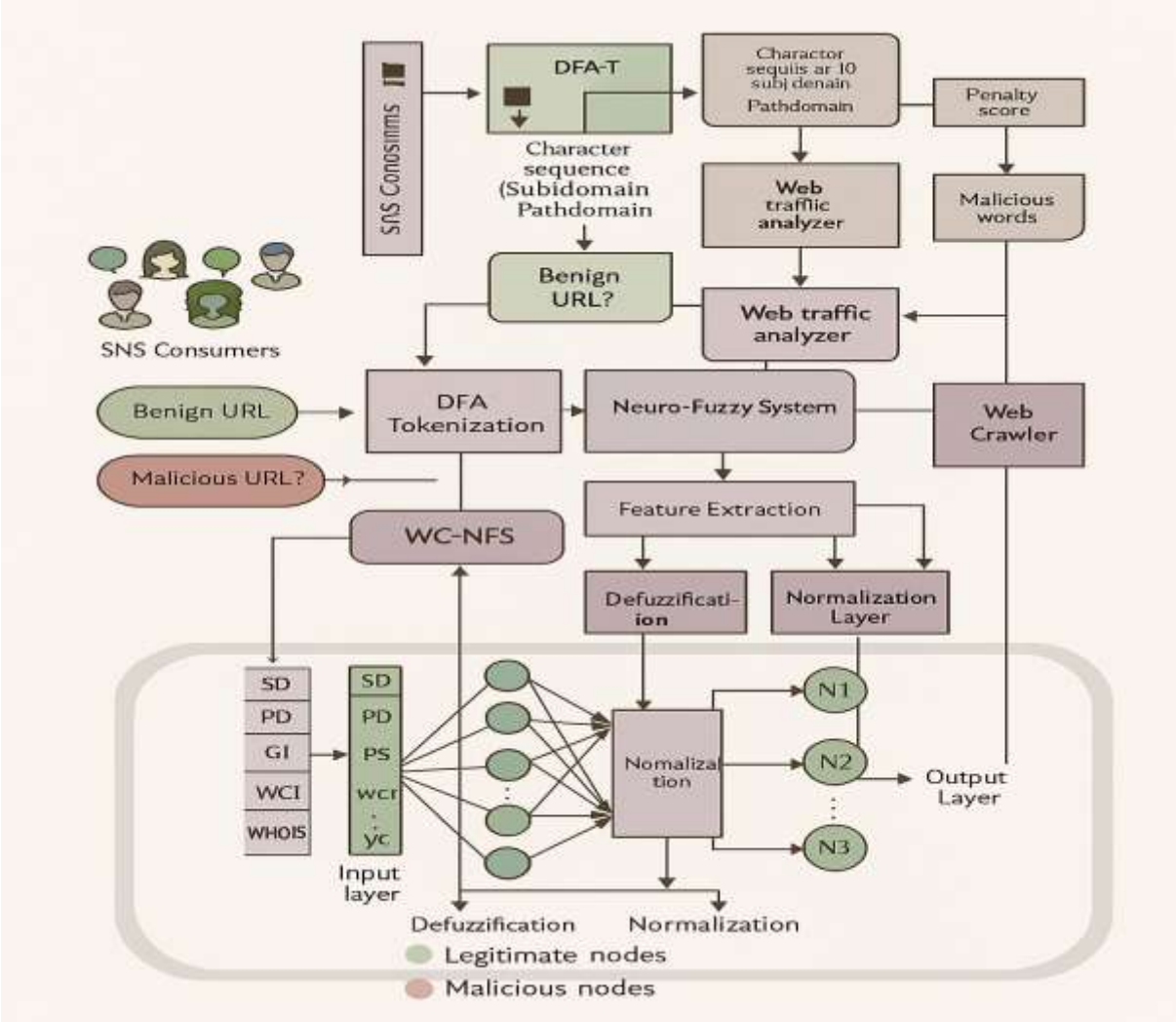


Figure 1. Overall workflow of SFD

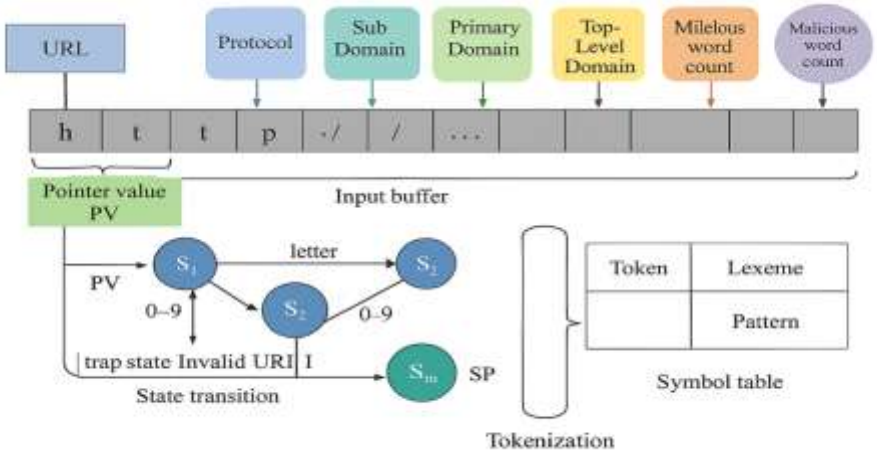


Figure12 Tokenization process S1 is starting state, S2and S3 are final state, S4 Invallic states, Sm malicious systems is Special character

10. Detection of Fraudulent Promotions in Online Social Networks

Online Social Networks (OSNs) have become prime targets for fraudulent promotions. Malicious actors exploit these platforms by posting deceptive advertisements. To address this, the Social Fraud Detection (SFD) model is proposed to identify and eliminate fraudulent marketing in **OSNs.**

10.1Overview of the SFD System

The SFD system classifies URLs shared on social media as safe or malicious using three major modules:

- **DFA Tokenization (DFA-T)**: Extracts and tokenizes URLs.
- **Web Crawler Index (WCI)**: Detects new malicious sites.
- **Web Classifier with Network and Feature Signals (WC-NFS)**: Integrates DFA-T and web crawling data for final URL classification.

FRAUD DETECTION PROCESS

10.2.1 DFA Tokenization (DFA-T)

URLs are parsed using a deterministic finite automaton, splitting them into components like subdomain, primary domain, and path. Tokens are processed for feature extraction.

10.2.2 Web Traffic Analyzer (WTA)

WTA calculates the Google Index and processes WHOIS data (registration/expiry dates) to detect suspicious websites. Malicious URLs often have recent or unstable registrations.

10.2.3 Web Crawlers

Web crawlers traverse user-shared URLs and related links. They calculate a Web Crawler Index (WCI) based on link traversal, where benign sites typically have a higher WCI than malicious ones.

7.2.4 Web Crawler-based Neuro-Fuzzy System (WC-NFS)

WC-NFS combines neural networks and fuzzy logic to classify URLs based on six features: Subdomain, Primary Domain, Penalty Score, WCI, Google Index, and WHOIS info. Features are normalized and fuzzified before classification.

EXPERIMENTAL RESULTS

11.1 Dataset

Datasets are collected from:

- **PhishTank** (anti-phishing data)
- **Malicious_n_nonmalicious URL** (400K labeled URLs)
- **Majestic Million CSV** (top ranked websites)

11.2 Environment Setup

About 33K URLs were tested with an 80:20, 70:30, and 60:40 benign:malicious split. DFA-T and WCI features were extracted and classified via WC-NFS, implemented in Python.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

11.3 Evaluation Metrics

The model was evaluated using **accuracy**, **sensitivity**, and **specificity**.

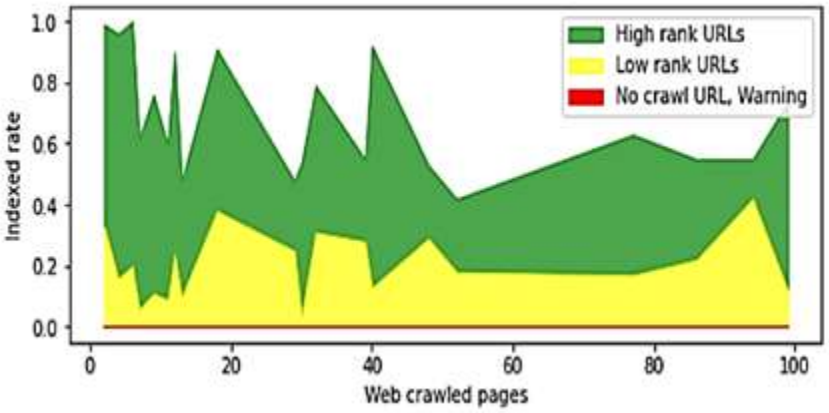


Figure 5.6 Web crawler Indexing

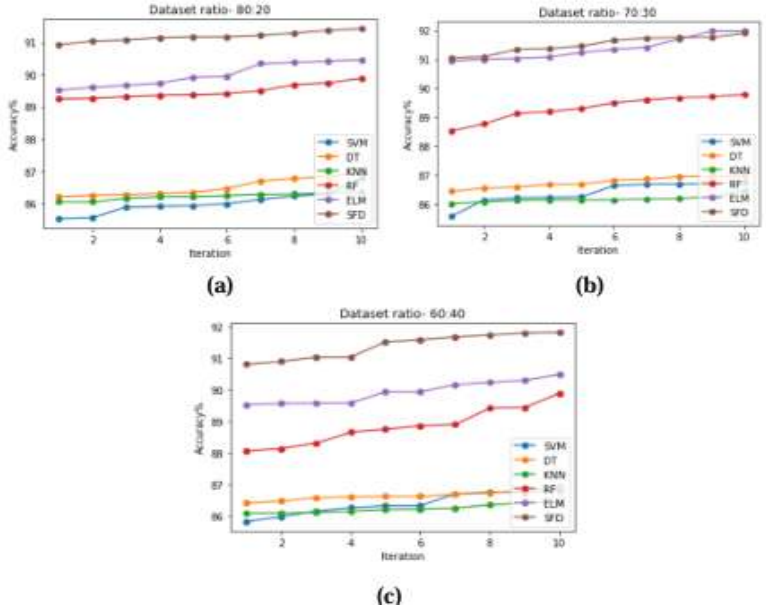


Figure 5.7 Accuracy evaluation of different ratios of benign and malicious URLS (a) 80:20,(b)70:30,(c)60:40

RESULTS

SFD achieveda **92.6% accuracy**, outperforming traditional classifiers like SVM, Decision Tree, KNN, Random Forest, and ELM. WCI analysis showed benign URLs generally have higher indexing values. Testing across various benign:malicious ratios confirmed SFD's robust performance.

CONCLUSION

The SFD model effectively detects fraudulent promotional posts on OSNs. By integrating DFA-T, WTA, and WC-NFS, it surpasses existing models in accuracy, sensitivity, and specificity, achieving a high classification accuracy of **92.6%**.

REFERENCES

[1]. Anderson, J., &Rainie, L. (2019). *The future of well-being in a tech-saturated world*. Pew Research Center. https://www.pewresearch.org
[2]. Cadwalladr, C., & Graham-Harrison, E. (2018, March 17). Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *The Guardian*. https://www.theguardian.com
[3]. Erguler, I. (2016). Achieving flatness: Selecting the number of honeywords in a password database without compromising security. *IEEE Transactions on Dependable and Secure Computing, 13*(2), 284–295. https://doi.org/10.1109/TDSC.2015.2415794
[4]. Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences, 80*(5), 973–993. https://doi.org/10.1016/j.jcss.2014.02.005
[5]. Juels, A., &Rivest, R. L. (2013). Honeywords: Making password-cracking detectable. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (pp. 145–160). https://doi.org/10.1145/2508859.2516671
[6]. Khandelwal, S. (2014, August 28). 5 Million Gmail usernames and passwords leaked online. *The Hacker News*. https://thehackernews.com
[7]. Kioon, Y., Park, J. H., & Lee, C. (2013). Securing password hashing with salt: A survey. *International Journal of Security and Its Applications, 7*(6), 55–66. https://doi.org/10.14257/ijsia.2013.7.6.06
[8]. Kelsey, J., Schneier, B., Wagner, D., & Hall, C. (2006). Secure applications of low-entropy keys. In *Information Security and Privacy* (pp. 121–134). Springer. https://doi.org/10.1007/3-540-36288-7_10
[9]. Mendelson, L., Wang, Y., & Hu, Y. (2020). On the effectiveness of password hashing and salting. *Journal of Cybersecurity, 6*(1), tyaa007. https://doi.org/10.1093/cybsec/tyaa007
[10]. Qiu, T., Zheng, K., & Li, B. (2020). Survey of authentication techniques for mobile devices. *IEEE Communications Surveys & Tutorials, 22*(3), 1565–1590. https://doi.org/10.1109/COMST.2020.2990750
[11]. Schneier, B. (2013). *Schneier on security*. Wiley.
[12]. Smith, A. (2020). *Social media use in 2020*. Pew Research Center. https://www.pewresearch.org
[13]. Wang, D., Zhang, Z., & Wang, P. (2019). A study of password security based on hashing and salting. *Computers & Security, 87*, 101571. https://doi.org/10.1016/j.cose.2019.101571
[14]. Ziegler, C., Gollmann, D., &Kiltz, E. (2014). On the security of password-based authentication. *IEEE Security & Privacy, 12*(6), 50–56. https://doi.org/10.1109/MSP.2014.124
[15]. Akshima, R., Gupta, G., &Bansal, D. (2019). Towards effective honeyword generation: Survey and research challenges. *ACM Computing Surveys (CSUR), 52*(6), 1–27. https://doi.org/10.1145/3360905
[16]. Wang, G., Zhang, T., Tang, H., & Huang, Y. (2021). Detecting insider threats in online social networks using LSTM-based anomaly detection. *Expert Systems with Applications, 165*, 113899. https://doi.org/10.1016/j.eswa.2020.113899